Center for Component Technology for Terascale Simulation Software

PI: R. Armstrong* (SNL), **Co-PIs:** D. Bernholdt (ORNL), L. Diachin (LLNL), D. Gannon (Indiana Univ.), J. Kohl (ORNL), G. Kumfert (LLNL), L. McInnes (ANL), J. Nieplocha (PNL), S. Parker (Univ. of Utah), and C. Rasmussen (LANL)

Summary

The Center for Component Technology for Terascale Simulation Software (CCTTSS) is dedicated to accelerating science by bringing a "plug and play" style of programming to high-performance computing. Through our programming model called the Common Component Architecture (CCA), we can dramatically reduce the time and effort required to compose independently created software libraries into new terascale applications. Major early-adopters of our technology include the application areas of combustion, quantum chemistry, and climate modeling, and efforts in many other areas are in early stages of development.

The CCTTSS center was formed around the Common Component Architecture (CCA) Forum, a grass-roots effort with the vision of bringing the software component paradigm to high-performance scientific computing, thereby reducing barriers to software reuse by allowing independently developed modules to be composed together to form complete simulations. This mode of programming is common in industry, but was unknown in scientific computing. Unlike commercial projects, the CCA addresses the challenges associated with maintaining high performance, working with a broad spectrum of programming languages and computer architectures, and helping to preserve DOE investments in legacy codes.

CCA Usage in Scientific Applications

The CCA is being used in an increasing number of scientific applications, where users are finding that it not only helps them to manage the burgeoning complexity of scientific software, but also allows them to take on scientific challenges that they would not have otherwise considered. For example, in computational quantum chemistry, CCA technology is facilitating collaboration, bringing state-of-the-art numerical algorithms from experts in optimization to problems such as molecular structure determination, which is a fundamental step in simulations ranging from combustion modeling to catalyst design. Furthermore, developers have explored algorithms with multiple levels of dynamic parallelism using CCA components

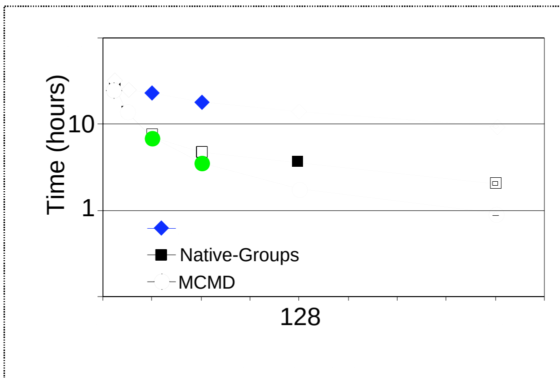

Figure 1. Multiple levels of parallelism via CCA components improve the scalability of Hessian evaluations within NWChem.

* 925-274-2740, rob@sandia.gov

in a multiple-component, multiple-data (MCMD) style. As shown in Figure 1, dynamic creation and management of subgroups have enabled the expression of multiple levels of parallelism, yielding significant performance improvements in NWChem. CCA-based applications are also being developed for climate modeling, combustion, nansoscience, biology, nuclear power plant simulation, and many others.

CCA Technology Development

The CCTTSS is addressing many research issues in order to create a practical environment for component-based high-performance scientific applications.

Frameworks: CCA frameworks, which are specialized for different computing environments, are tools that compose components into full applications. Ccaffeine and SCIRun2 support traditional parallel computing, while XCAT, SCIRun2, and Legion-CCA support distributed computing. A new framework, DCA, allows multiple traditional parallel applications to execute as a single application.

Language Interoperability: Because application scientists use a variety of programming languages, the CCA has been designed to be language neutral. Our Scientific Interface Definition Language (SIDL) and associated compiler, Babel, allow component interfaces to be written and used from any of C, C++, Python, and Fortran with minimal performance costs.

Scientific Components: As a part of its mission, the CCTTSS has developed production components that are used in scientific applications as well as prototype components that aid in teaching CCA concepts. These freely available components include tools for mesh management, discretization, linear algebra, integration, optimization, parallel data description, parallel data redistribution, visualization, and performance evaluation. We are also collaborating with the APDEC[1], TSTT[2] and TOPS[3] SciDAC centers to define common interfaces for mesh-based scientific data management and linear solvers.

Outreach and Usability Enhancements

Much of the CCA's success thus far has been due to the outreach done by the CCTTSS, including numerous tutorials and papers at major scientific meetings, as well as journal articles and book chapters. We have also introduced new tools that aid users in the transition from a static application code to reusable components.

CCA into the Future

To date, SciDAC funding has accelerated the development of CCA technology and its insertion into applications. Because at this point the CCA has a proven specification and methodology, we will now focus on improving component writers' and users' experiences, with emphasis on integrating legacy code into large-scale multi-physics applications. The CCA has made significant headway on its goal of changing scientific software development from many small, individual efforts into larger, community-oriented projects. In the coming years we will seek to make this community model the norm through mature software that helps to streamline and expedite the transition.

For further information on this subject contact:
Robert Armstrong (PI)
Sandia National Laboratories
Livermore, CA 94551
Phone: 925-294-2470
E-mail: rob@sandia.gov
http://www.cca-forum

---

[1] *Algorithmic and Software Frameworks for Applied Partial Differential Equations*, PI: P. Collella, http://davis.lbl.gov/APDEC
[2] *Terascale Simulation Tools and Technologies Center*, PIs: J. Glimm, D. Brown, and L. F. Diachin, http://www.tstt-scidac.org
[3] *Terascale Optimal PDE Simulations Center*, PI: D. Keyes, http://tops-scidac.org